Format Diskette (FORMAT)

(V mode only)

FORMAT [side:] [/D](cr)

FORMAT is an external command that prepares floppy diskettes in VICTOR format to receive data. As it formats the storage space on a diskette, FORMAT erases any previous files. You can load the FORMAT program, like the DISKCOPY program, in either of two ways—by typing only the program name, or by typing a full command line. Like DISKCOPY, FORMAT has a full-screen mode that enables you to process more than one diskette without reloading the program for each one.

If you enter:

format(cr)

the program loads into memory and displays a screen that asks you to:

- Select the drive (left or right) containing the diskette to be formatted.
- Specify an optional Volume ID (an on-disk label of up to 11 alphanumeric characters) to identify the diskette.
- Select double-sided format if your computer has double-sided drives.

To make selections, use the cursor keys on the right side of the keyboard. Then use the function keys at the top of your keyboard to start or quit the FORMAT process. The bottom line of the screen shows the current value of the function keys. For example, you can get onscreen HELP by pressing F7.

If you include the drive side in your FORMAT command line, the program formats the diskette on the side you name and then exits to the operating system. For example, you can type:

format /d right(cr)

The FORMAT program displays its one-line sign-on message and immediately begins formatting the diskette in the right drive (drive B). FORMAT tells you when the process is complete, and then redisplays the system prompt. The drive side can be abbreviated to L (left) or R (right).

The /D switch with the FORMAT command formats a double-sided diskette. If you do not specify /D, your diskette is formatted as single-sided.

Go To Label (GOTO)

GOTO label(cr)

GOTO is an internal command for batch files. Batch processing continues with the command on the line below the specified label. GOTO is issued as a separate command except when used with the IF command. IF uses GOTO on the same command line as IF.

label
is a string of up to 8 alphanumeric characters. The label in the GOTO command must match a label in the batch file. The label in the batch file must be preceded by a colon, telling MS-DOS to ignore this line.

The line containing the label can precede or follow the GOTO command in the batch file. When the label precedes GOTO, the batch file goes into a loop. When the label follows GOTO, batch processing skips the intervening commands. If the batch file contains a GOTO command that does not include a label, the batch file ends.

In the following example, the double = tells MS-DOS that this is an "equal to" symbol, not an assignment command, such as SET PATH = A:\COMDIR.

```
copy con private.bat(cr)
echo off(cr)
if %1 == security goto greet(cr)
echo Access denied(cr)
goto end(cr)
:greet(cr)
echo Welcome! Enter your name!(cr)
:end(cr)
^Z(cr)
```

To process PRIVATE.BAT and to substitute the word SECURITY for the replaceable parameter %1, type:

```
private security(cr)
```

MS-DOS displays:

```
A>echo off
Welcome! Enter your name!
```

If you specify any word other than SECURITY, MS-DOS displays:

```
A>echo off
Access denied
```

The ECHO OFF command stops the display of the batch commands during batch file processing. MS-DOS replaces %1 with the name you entered when you specified the batch file. MS-DOS compares the name to the one in the IF command. When there is a match, the welcome greeting and the prompt message for a name are displayed. If there is

no match, the next command processed is the ACCESS DENIED message. MS-DOS then goes to the command following the label :END, which is the end-of-file marker (^Z).

Note: The labels are not displayed during batch file processing because MS-DOS ignores lines that start with a colon. You can add non-displayed comments to a batch file by starting each comment line with a colon.

Print Graphics Display Screen (GRAPHICS)

(I mode only)

GRAPHICS(cr)

GRAPHICS is an external command that allows the contents of a graphics display screen to be printed on printers that support the printing of individual dots.

Enter the GRAPHICS command after loading MS-DOS. Any time you want a printout of the screen, press SHIFT-PRT SC.

If the screen is in text mode, the printout will be in text mode.

If the screen is in the 320 × 200 color graphics mode, each screen dot is printed in one of four shades of gray.

If the screen is in the 640 × 200 color graphics mode, the screen is printed sideways on the paper.

Printing speed is dependent on the speed of the printer being used.

Display Command History (HISTORY)

HISTORY [n](cr)

HISTORY is an internal command that captures console input. HISTORY remembers the commands you have entered. With HISTORY you can display the command history, and you can select any command from the history to re-enter into the MS-DOS command-line template. Then, using the function keys as described in Chapter 5, you can edit and/or re-enter that command without retyping the entire line.

MS-DOS can store a history of any number of command lines, up to 256 characters. Typically, HISTORY retains from 10 to 25 commands. HISTORY commands themselves, however, are not stored in the command history. You can abbreviate HISTORY to HI.

Entering HISTORY with no parameters displays the command history onscreen, like this:


```
n.  <oldest line in HISTORY>
.     .     .     .
.     .     .     .
2.  <command entered before #1>
1.  <command entered before this HISTORY command>
```

Commands too long to fit the display line wrap to the next line.

n
: is the number of a command stored in the history. The nth command from the history is placed into the command-line template, where you can edit and/or execute it as shown in Chapter 5.

For example, entering:

hi 5(cr)

puts the fifth line from the command history into the command-line template. If you want to execute that command without changing it, press F3 (Copy Template) in I mode or F7 (Copy Template) in V mode.

HISTORY 1 refers to the last command entered before any HISTORY command(s). HISTORY 2 is always the non-HISTORY command before that.

Reload Command Processor (ICOMMAND/VCOMMAND)

ICOMMAND [path][device] [/D] [/P] [/C string](cr)

VCOMMAND [path][device] [/D] [/P] [/C string](cr)

ICOMMAND/VCOMMAND is an external command that invokes the command processor, ICOMMAND.COM or VCOMMAND.COM depending on the mode you are using: I mode or V mode. When you enter ICOMMAND or VCOMMAND, MS-DOS reloads the appropriate command processor.

You can use ICOMMAND or VCOMMAND interactively by entering the command from the keyboard at the system prompt, or you can use ICOMMAND or VCOMMAND in batch files to reload the command processor.

path
is the path through the directory system to the file containing the command processor. The path includes the drive name and subdirectory names where necessary. No path is necessary if the file resides in the root directory on the default drive.

device
is the name of the TTY device. If you do not specify a device, MS-DOS uses CON (by default, the keyboard and screen).

/D
omits prompts for date and time when the command processor reloads. If you do not specify /D, you are prompted for the date and time.

/P
places ICOMMAND.COM/VCOMMAND.COM in memory until you reload the operating system. /P also tells the command processor to search for an AUTOEXEC.BAT file and to prompt you for the date and time as it loads into memory. If /P is omitted, ICOMMAND.COM or VCOMMAND.COM is a transient command; you can exit the current command processor loaded in memory and return to the previously loaded command processor.

/C string
submits a string for the command processor to execute after it loads into memory. ICOMMAND.COM/VCOMMAND.COM executes this single command string and then exits. The /C switch overrules the /P switch. If present, the /C switch must be the last switch. All text following /C is passed to the command processor.

If Condition (IF–EXIST)

IF [NOT] string1 == string2 command(cr)

or

IF [NOT] EXIST [path]filename.ext command(cr)

or

IF [NOT] ERRORLEVEL n command(cr)

IF is an internal command for batch files. If a condition is true, MS-DOS processes the command specified in the IF command line. Otherwise, MS-DOS processes the next command in the batch file.

NOT
processes the command when the IF NOT condition is true.

string1 == string2
compares one alphanumeric string of characters (string1) with another string of alphanumeric characters (string2). To MS-DOS, == means "equal to." If string1 matches string2, the command is processed. String1 is usually a replaceable parameter, such as %1. You cannot use command delimiters, such as commas, within the string parameters.

For example, this batch file uses the IF command and the string parameters to verify system users:

```
copy con users.bat(cr)
echo off(cr)
if %1 == George goto label(cr)
if %1 == Jack goto label(cr)
if %1 == Mary goto label(cr)
echo You must be an approved user to operate the system.(cr)
goto end
:label(cr)
echo Welcome! Enter your command.(cr)
:end
^Z(cr)
```

When you specify USERS and a name to be processed, MS-DOS replaces the %1 parameter in the first IF command with the specified name. MS-DOS compares that name (string1) to the string of characters in string2. If they are equal, MS-DOS displays:

```
Welcome! Enter your command.
```

If there is not a match in the first command, the next command is processed. If MS-DOS processes the three IF commands in the file without a match, MS-DOS displays:

```
You must be an approved user to operate the system.
```

EXIST [path]filename.ext

tells MS-DOS to look for the specified file in the specified path. If you do not include a path, MS-DOS searches the current directory on the default drive. If the file is found, the command is processed. For example, this command in a batch file:

if not exist \mgr\jobs echo Can't find \mgr\jobs.(cr)

searches for the file JOBS in the MGR subdirectory on the disk in the default drive. If JOBS is found, MS-DOS does not process the ECHO command. If \MGR\JOBS does not exist, MS-DOS displays:

```
Can't find \mgr\jobs.
```

ERRORLEVEL n
tests for a program failure. This parameter tells MS-DOS to check for an error level number, where *n* is the specified number. The next command is processed if the previous program run by MS-DOS had an exit code the same as *n* or higher.

command
can be any of the MS-DOS commands. The most common command used with IF is GOTO.

List Files (LS)

LS [options] [files or directories](cr)

LS.EXE is an external command that produces a sorted listing in various directory formats of a file or list of files. In the LS listing, names of subdirectories are followed by a \ (such as MGR1\).

Options can be any of the following switches:

/B
sorts backwards (Z to A).

/E
sorts output by file extension.

/G
lists files "toGether." Files are not separated by directory.

/L
lists the directory entries in long format. The filename, file attribute, date, time, and size are listed for each file. Long-format listings are not sorted.

/R
recursively lists all files in subdirectories of any directory specified. You can use this switch to list the contents of all directories on a drive. LS /R lists each subdirectory followed by all the files in that directory.

/T
sorts output by date and time of creation, rather than alphabetically.

/1
lists only one filename per line.

files or directories
is a list of one or more filenames and/or directories. If this parameter is omitted, LS lists the current directory.

The following example alphabetically lists the files and subdirectories in the current directory:

ls(cr)

The next example lists, in long format, the files in the \TEMP directory (and its subdirectories) on drive B, and any file with .ASM extension in the working directory:

ls /rl b:\temp *.asm(cr)

The listing on your screen will be similar to the following:

```
\temp:
a----w    1430   24-SEP-84   12:57 BLDTYPE.ASM
a----w     616   20-MAR-85   10:48 BRITE.ASM
a----w    6247   02-FEB-85   12:37 DTIME.ASM
a----w   60491   10-OCT-84   10:48 HMANAGER.ASM
a----w     229   09-FEB-85   17:44 OMANAGER.ASM
5 files.
```

Make Directory (MKDIR)

MKDIR [path](cr)

MKDIR is an internal command that makes a new subdirectory beneath the root directory, which MS-DOS created when you formatted your disk. You can abbreviate MKDIR to MD. See Chapter 3 for more information on creating and using subdirectories.

path
is the path through the subdirectory hierarchy to the subdirectory you want to create. The last directory name in the path is the name of the new subdirectory. MS-DOS can create a subdirectory only if a directory above it exists. An initial \ always indicates the root directory. An initial directory name without a \ indicates a subdirectory below the current directory.

For example, if MS-DOS is at the root directory and you enter:


```
mkdir mgr1.nam(cr)
mkdir mgr2.nam(cr)
mkdir mgr3.nam(cr)
```

MS-DOS creates these three subdirectories at the first level beneath the root directory. Each subdirectory is listed in the root directory. You can verify this by entering the DIR command. MS-DOS displays:

```
Volume in drive A is June 8 1985
Directory of A:\

MGR1      NAM <DIR>      6-30-85   9:00a
MGR2      NAM <DIR>      6-30-85   9:00a
MGR3      NAM <DIR>      6-30-85   9:01a
       3 File(s)    370683 bytes free
```

"Directory of A:\" indicates the root directory. Note that subdirectories are displayed with <DIR>; files are not displayed with this attribute.

Once a directory exists, you can make a subdirectory below it. For example, this command creates the subdirectory JOBS below the subdirectory MGR1.NAM:

```
mkdir mgr1.nam\jobs(cr)
```

You can enter a directory name without a \ in a MKDIR command to add a subdirectory to the directory you are currently using. For example, if you are working in the MGR1.NAM directory, instead of issuing the preceding command, you can specify:

```
mkdir jobs(cr)
```

Once the JOBS subdirectory has been created, you can create a subdirectory below it:

```
mkdir mgr1.nam\jobs\benefits(cr)
```


or, if the current directory is JOBS, enter:

```
mkdir benefits(cr)
```

You can make as many subdirectories as you want depending on the amount of space on your disk. Subdirectory names can be 1 to 8 characters followed by an extension of 1 to 3 characters. The valid characters for filenames are valid for directory names. You can duplicate file or directory names as long as they are in separate directories. When you are creating subdirectories, do not use more than 63 characters in the command, including the \.

When you finish making your directory hierarchy, it is a good idea to enter the CHDIR \ command. This command returns you to the root directory. You should always return to the root directory so that you do not accidentally create an entry in the wrong subdirectory.

Modify Console (MODCON)

(V mode only)

MODCON sourcefile [savefile](cr)

MODCON.EXE is an external command for modifying the "soft" keyboard and character set used by the console. A soft keyboard is a part of the operating system that tells the computer what character and code to generate for each keystroke. With MODCON you can select a new keyboard file and/or character set file. You can also save the current character set(s), and then restore them later.

sourcefile
is the file or files containing the sets to be made active. Follow these rules for the source filename:

- If the source file is in a subdirectory, include the path through the directory system to the file (see Chapter 3).
- To select only a keyboard file or only a character set file, include the .KB or .CHR extension for the source file.
- If the source filename has no extension, MODCON activates both the .KB and the .CHR files with that filename.
- If you enter an asterisk (*), no new sets are made active and the configuration remains unchanged.

savefile
is the file or files used to save the current keyboard table and/or character set. Follow these rules for the save filename:

- If you want to save the existing file(s) in a subdirectory, include the path through the directory system to the file or files.
- If you are saving only a keyboard file or only a character set file, the saved file should have .KB or .CHR extension, respectively.
- To save both keyboard and character set files, omit the extension for the saved filename.

Note: The extension for the source file is independent of the extension for the saved file.

Here are some example command lines:

modcon b:\dir3\test.chr \chr\save.chr(cr)

This command modifies the character set, but does not affect the keyboard. The new character set is TEST.CHR in the \DIR3 subdirectory on drive B. The current character set is saved in SAVE.CHR in the \CHR subdirectory on the default drive.

modcon \chr\test.kb(cr)

This command selects a new keyboard file, called TEST.KB in the \CHR subdirectory on the default drive, but does not save the current keyboard file.

modcon germ01 g02save(cr)

This command selects GERM01.KB and GERM01.CHR as the new keyboard and character set. The current keyboard table and character set are saved in G02SAVE.KB and G02SAVE.CHR on the default drive.

modcon germ01 germ01(cr)

This command activates the GERM01.KB and GERM01.CHR files on the default drive. The save operation overwrites the original GERM01.KB and GERM01.CHR files with the current character set and keyboard.

modcon aust01(cr)

This command makes AUST01.KB and AUST01.CHR the active keyboard table and character set. The original keyboard table and character set are not saved.

modcon * brit01.kb(cr)

This command saves the active keyboard set in the file BRIT01.KB on the default drive, but leaves it as the active set. The character set is not affected.

modcon french.kb b:swedish.chr(cr)

This command saves the current character set in B:SWEDISH.CHR and leaves that character set active. It also uses the keyboard from FRENCH.KB on the default drive, overwriting the current keyboard table in memory without saving it.

Character Sets and Keyboard Files

To use MODCON, you must have on disk the keyboard or character set files that you want to use in your configuration.

Character sets are obtained by:

- Selecting a set provided on the system selection (SYSGEN) diskette included in the *Applications Programmer's Tool Kit II.*
- Selecting a graphics character set provided with CHARGRAF in the *Graphics Tool Kit II.*
- Modifying the current set or any available set using the EFONT character-font editor included in the *Graphics Tool Kit II* and in the *Applications Programmer's Tool Kit II.*

Keyboard tables are obtained by:

- Selecting a table provided on the system selection (SYSGEN) diskette included in the *Applications Programmer's Tool Kit II.*
- Modifying the current table or any available table using the keyboard table editor KEYGEN included in the *Graphics Tool Kit II* and in the *Applications Programmer's Tool Kit II.*

Note: You can purchase the preceding tool kit packages from your dealer.

Set Operation Modes (MODE)

MODE is an external command that sets operation modes for the video display and for printer or serial ports. There are four options for this command. Each has its own function.

MODE LPT #:[n][,[m][,P]](cr)

Sets mode parameters for a printer

MODE n(cr)

Sets mode parameters for a video display, for I mode only

MODE COMn:baud[,parity[,databits[,stopbits[,P]]]](cr)

Sets mode parameters for a serial port

MODE LPT #:=COMn(cr)

Redirects parallel printer output to a serial port

1. Set Mode Parameters for a Printer

With this option you can set the operation mode for a printer. The syntax is:

MODE LPT#:[n][,[m][,P]](cr)

#	Establishes the printer number (1, 2, or 3).
n	Establishes the number of characters per line (80 or 132).
m	Establishes the number of vertical lines per inch (6 or 8).
[,P]	Establishes continuous retry on timeout errors.

If you do not specify a value for m or n, or if the value you specify is invalid, then the mode for that operation does not change. You can stop continuous retries by pressing CTRL-BREAK. If you do not want timeout errors to continuously retry, do not specify the P parameter.

For example, if you want to set line printer number 2 for continuous retry and 132 columns with 8 line per inch spacing, enter the following command:

```
mode lpt2:132,8,p(cr)
```

2. Set Mode Parameters for a Video Display (I mode only)

Option 2 sets the operation mode of the video display(s). The syntax is:

```
MODE n(cr)
```

n Establishes the type of display (40, 80, BW40, BW80, CO40, CO80, or MONO). If the color dipswitch is set, specifying MONO does nothing to the display. You must physically set the dip switch to change the type of display from monochrome to color.

40 = 40 character display width (color/graphics display mode)
80 = 80 character display width (color/graphics display mode)
BW40 = Black and white, 40 character display width
BW80 = Black and white, 80 character display width
CO40 = Color, 40 character display width
CO80 = Color, 80 character display width
CO320 = Color/Graphics, 320 × 200 display
BW320 = Color/Graphics, 320 × 200 display
BW640 = Color/Graphics, 640 × 200 display
MONO = Monochrome, 80 character display width

3. Set Mode Parameters for Serial Communications Ports

This option allows you to set the operation mode for serial ports. The syntax is:

MODE COMn:baud[,parity[,databits[,stopbits[,P]]]](cr)

n	Establishes the serial port number (1 or 2).
baud	Establishes the baud rate (110, 150, 300, 600, 1200, 2400, 4800, or 9600). Only the first two digits are required (for example, 11, 15, 30, 60, and so on).
parity	Establishes the parity mode (N = none, O = odd, E = even). The default is E (even).
databits	Establishes the number of bits in a serial byte (7 or 8). The default is 7.
stopbits	Establishes the number of bits signaling the end of a serial byte (1 or 2). If the baud rate is 110, the default is 2; otherwise, it is 1.
[,P]	This parameter causes continuous retries to occur on timeouts. This is useful when you use a serial printer that cannot keep up with the serial rate. Press CTRL-BREAK to stop the retry loop. If you do not want the timeout errors to continuously retry, reinitialize the serial port with the P parameter.

You must specify values for the serial port number and the baud rate; if you omit values for the other parameters, MODE assigns the default values. When you do not enter a parameter, enter a comma for the value omitted.

For example, to set up serial port number 2 for 9600 baud, even parity, 8 databits, 2 stopbits, and continuous retries, enter the following command:

mode com2:96,e,8,2,p(cr)

To use the default values for this command, enter the following:

```
mode com1:12,,,,p(cr)
```

4. Redirect Parallel Printer Output to a Serial Port

This option redirects output for a parallel printer to an asynchronous communications adapter. With this option, you can redirect output for the logical printers to the serial I/O ports. Before you use this option, you must use Option 3 to initialize the serial port. The syntax for Option 4 is:

```
MODE LPT#: = COMn(cr)
```

Establishes the printer number (1, 2, or 3).

n Establishes the serial port number.

The following command causes all outputs to line printer number 3 to be redirected to serial port number 2:


```
mode lpt3: = com2(cr)
```

To disable the redirection for the printer number 3, enter:

```
mode lpt3:(cr)
```

More Screen (MORE)

MORE(cr)

MORE.COM is an external filter command that passes through the data in a file and displays one screen of data. At the bottom of the screen, this message is displayed:

```
-- More --
```

Press Return or the space bar to display another screen of data. The MORE message displays with each new screenful of data until MS-DOS reads the entire file.

MORE is used mostly with other commands to pipe data (see Chapter 5.3). For example, you can display a directory one screen at a time:

dir | more(cr)

The following command uses MORE in a pipe and filter operation:

dir | find "DIR" | more(cr)

The current directory is piped as input to the FIND command. FIND searches the directory for all DIR entries. The output of FIND is piped as input to MORE. The MORE command stops the directory display when the screen is full. When you press the Return key, another screen of directory entries is displayed.

For more information on filtering data, see Chapter 5.2.

Move or Rename Files (MV)

MV [path]filename.ext [path]filename.ext(cr)

or

MV [path]filename.ext... path(cr)

MV.EXE is an external command that moves files. If you name two files in an MV command, the first file overwrites the second file, and the original file is deleted. If you name a file and a directory, the file is copied into the directory, and the original file is deleted. If you name multiple files and a directory, all the files are moved into the last named directory.

path
: is the path through the subdirectory system to the file(s) you want to move, or to the file or directory where you want to move the file(s).

filename
: In the first parameter, filename is the name of the file(s) you want to move. If the second parameter names a file, the first file is copied into the second file, and the original file is deleted. If the two files are on the same drive, MV renames the first file as the second filename, and deletes the second file if it already exists.

You can use wild-card characters in the first filename if you are moving files into a directory. If you use wild-card characters in the first filename of an MV command moving files into another file, however, MV moves only the first matching file into the destination file.

Then MV displays an error message like this:

```
I already moved d:FILENAME.1 to d:DESTFILE
and will not now move d:FILENAME.3 to d:DESTFILE
```

The following example moves all .EXE files to the \BIN directory on drive B:

```
A>mv *.exe b:\bin(cr)
     mv MV.EXE          to (directory) b:\bin
     mv CONCAT.EXE      to (directory) b:\bin
     mv FORMAT.EXE      to (directory) b:\bin
     .    .    .        .    .        .  .
     .    .    .        .    .        .  .
```

This MV command renames MEMO.MEM to SEPT28.MEM:

mv memo.mem sept28.mem(cr)

You can refer to any number of drives in an MV command. For example, this command moves all .EXE files and all files with names starting with SAVE on drive A to the \BACKUP directory on drive B:

mv a:*.exe a:save*.* b:\backup(cr)

Path to Command File (PATH)

PATH [path][;path]...(cr)

PATH is an internal command that searches first the current directory then the other specified directories for command or batch files (.COM, .EXE, or .BAT files **only**). PATH allows you to load external commands and batch files from a subdirectory without typing the full path to the subdirectory each time. PATH without parameters displays the previously specified PATH command.

For example, if you enter this command:

path \cmd(cr)

and then enter:

path(cr)

MS-DOS displays:

```
PATH=\CMD
```

If a PATH command was not previously specified, MS-DOS displays: "No Path".

;

used with PATH and no other parameters tells MS-DOS to search only the current directory for a command or batch file. A semicolon between directory names in the path specifies more than one path (indicated by the \s) for the command or batch file search.

path
is the path through the subdirectory system to the subdirectory containing the external commands and batch files.

Suppose BIN is the first-level subdirectory and contains all the external commands. You can specify:

path \bin(cr)

Then you can enter any external command without specifying the \BIN subdirectory, regardless of the directory you are using.

If you want MS-DOS to search more than one drive or directory for a command or batch file, you can specify:

path a:\bin;a:\bin\utils;c:\pascal\programs;d:\(cr)

MS-DOS searches the current directory first, then the PATH directories in the order that they were specified. In the above example, the directory A:\BIN is searched before A:\BIN\UTILS. The first .COM, .EXE, or .BAT file found whose name matches the command entered will be loaded.

If your PATH command contains an invalid delimiter or parameter, such as the wrong drive name, MS-DOS does not detect it until it looks for a command or batch file. Then MS-DOS displays either or both of the following error messages:

```
Invalid drive in search path
Bad command or file name
```

Pause Batch Process (PAUSE)

PAUSE [comment](cr)

PAUSE is an internal command for batch files that suspends batch execution and displays the message: "Strike any key when ready...". Pressing any key except CTRL-C resumes execution of the batch file. Pressing CTRL-C displays:

```
Abort batch job (Y/N)?
```

If you press Y, the remainder of the batch file is not executed, and the MS-DOS command prompt is redisplayed.

At each PAUSE in the file, MS-DOS stops. You can then decide if you want to end processing or do some other action, such as changing diskettes.

comment
is any string of characters (121 maximum) that you specify. The characters are displayed with the "Strike any key when ready..." message.

For example, this command stops batch file execution:

pause Insert a diskette in drive b(cr)

MS-DOS displays:

```
Insert a diskette in drive b
Strike any key when ready...
```

Print Queued Files (PRINT)

PRINT [drivename:]filename.ext [/C] [/P] [/T] ...(cr)

PRINT is an external command that puts files in a print queue and prints them while MS-DOS is doing other tasks. Only files from the current directory can be put in the print queue.

You cannot specify a path with the PRINT command. Instead, use the CHDIR command to move to the directory that contains the file you want to print. PRINT without any parameters displays the names of the files in the queue.

When you specify PRINT for the first time after MS-DOS starts, MS-DOS displays this message:

```
Name of list device [PRN]:
```

The list device is the printer. Specify the valid assignment name for your list device, such as COM1, COM2, or LPT1 (see the CLST command in this chapter for more information). If you press the Return key, MS-DOS uses PRN as the list device. PRN is a logical device name equivalent to LST.

After you give the device name for your printer, the screen displays:

```
Resident part of PRINT installed
```

If you give an invalid assignment name for the device, MS-DOS displays:

```
List output is not assigned to a device
```

Be sure that the list device is attached to your computer.

The list device cannot be used when PRINT is being processed. MS-DOS displays an error message until the files are all printed or printing is terminated with the /T parameter.

drivename
is the name of the drive containing the directory of files that you are currently using. If you do not specify a drive, MS-DOS uses the default drive. Do not remove the disk from the drive while the files are being printed. If MS-DOS tries to access the file and the drive is not ready, MS-DOS displays:


```
Drive not ready
```

If MS-DOS finds a disk error, the file being printed is canceled. The printer types a disk error message, advances to the next page, and the printer alarm sounds. MS-DOS then prints the rest of the files in the print queue.

filename
is the name of the file(s) that you want to print. MS-DOS queues and prints the files in the order you specify in the command. You can specify a maximum of 10 files.

If you enter PRINT and the name of the file(s) without any other parameters, for example:

```
print chapt1.doc chapt2.doc chapt3.doc(cr)
```

MS-DOS queues and prints the files. MS-DOS displays the name of the drive and the name of the first file being printed:

```
A:CHAPT1      .DOC is currently being printed
```

The names of the remaining files in the queue are also displayed:

```
CHAPT2      .DOC is in queue
CHAPT3      .DOC is in queue
```

MS-DOS prints a file, moves to the next page, then prints the next file. Tabs are expanded with blanks to the next eighth column.

Wild-card characters can be used in the filename and its extension (see Chapter 2.3).

/C

cancels the named file(s). The filename preceding the /C and subsequent filenames are canceled until MS-DOS finds a /P in the command line. For example, the following command cancels CHAPT1, CHAPT2, and CHAPT3 from the print queue and prints CHAPT4.DOC:

```
print chapt1.doc /c chapt2.doc chapt3.doc chapt4.doc /p(cr)
```

This command removes all CHAPT?.??? files from the print queue:

```
print chapt?.* /c(cr)
```

If there are no other files in the print queue and a file is printing—for example, MORE.TXT—and you enter:

print more.txt /c(cr)

MS-DOS prints:

```
A:MORE      .TXT Canceled by operator
```

The printer advances one page, and the printer alarm sounds once. MS-DOS then displays on your screen:

```
PRINT queue is empty
```


If you specify:

print more.txt echo.txt(cr)

MS-DOS displays:

```
A:MORE      .TXT is currently being printed
ECHO        .TXT is in queue
```

If you then type:

```
print more.txt /c(cr)
```

while the MORE file is printing, MS-DOS:

- Stops printing the MORE file
- Prints the operator-canceled message
- Advances the paper one page
- Sounds the printer alarm
- Prints the next file in the queue
- Displays:

```
A:ECHO     .TXT is currently being printed
```

/P

prints the named file(s). The file preceding the /P and subsequent files are printed until MS-DOS finds a /C in the command line. For example, this command prints CHAPT5 and CHAPT6, cancels CHAPT7, and prints CHAPT8:

```
print chapt5.doc /p chapt6.doc chapt7.doc /c chapt8.doc /p(cr)
```

/P is the default if you specify only PRINT and the filename.

/T

deletes all queued files from the print queue. When you use /T, do not specify any other parameters. For example, this command empties the print queue:

```
print /t(cr)
```

MS-DOS displays:

```
PRINT queue is empty
```

MS-DOS prints:

```
All files canceled by operator
```

MS-DOS Prompt (PROMPT)

PROMPT [prompt-text](cr)

PROMPT is an internal command that changes the MS-DOS default prompt. PROMPT without parameters designates the default prompt to be A>.

prompt-text
represents alphanumeric character(s) that you specify.

For example, you can specify the prompt as COMMAND by typing:

prompt COMMAND(cr)

MS-DOS displays the new system prompt instead of the A> prompt:

```
COMMAND
```

If you type the following, the A > prompt returns:

```
COMMANDprompt(cr)
A>_
```

The next example changes the prompt from A > to READY, and then reinstates the A > prompt:

```
A>prompt READY(cr)
READYprompt(cr)
A>_
```

You can insert special characters in the prompt by specifying $c for the prompt-text, where c is:

$	The $ character
t	The current time (set when MS-DOS was loaded or by the last TIME command)
d	The current date (set when MS-DOS was loaded or by the last DATE command)
p	The name of the current directory of the default drive
v	The MS-DOS version number
m	The current operating mode (V or I)
n	The default drive
g	The > character
l	The < character

b The | character

s A leading space only

e The ESC character

_ Carriage return/line feed, which returns to the beginning of the next line on the screen

The special characters inserted with $ must be lowercase. If you specify $ and any other character than the ones described, MS-DOS does not process the command.

To use the current time and date as a system prompt, type:

```
prompt TIME = $t$_DATE = $d(cr)
```

MS-DOS displays, as a two-line command prompt, the current time and date:

```
TIME=10:54:30.50
DATE=Mon 7-04-85
```

Recover Files (RECOVER)

RECOVER [path]filename.ext(cr)

or

RECOVER drivename:(cr)

RECOVER is an external command. Use this command when MS-DOS cannot read a file because of a defective disk. If a file or a diskette directory has a bad sector, this command recovers the file or the entire disk. If there is a bad sector in the directory, this command recovers all the files on the disk.

path
is the path through the subdirectory system to the file you want to recover.

filename.ext
is the file that has the bad sector. The file is read sector by sector from the disk. MS-DOS marks the bad sector, skips it, and allocates it to a system table. The data in the bad sector is not recovered. MS-DOS does not make any more file allocations to that sector.

The RECOVER command and the drivename without any parameters recovers all the files on the disk. MS-DOS scans the File Allocation Table (FAT) for chains of allocation units (the amount of space used on a disk for a file). MS-DOS creates a new root directory. Each entry in the root directory is renamed as:

FILEnnnn.REC

where *nnnn* is a sequential number starting with 0001. Each FILEnnnn.REC points to the beginning of an allocation chain.

If the root directory cannot contain all of the allocation unit chains, RECOVER displays a message. The unrecovered chains are left in the File Allocation Table (FAT). Reissue RECOVER and the drive name when you have made more room in the root directory.

Remark (REM)

REM comment(cr)

REM is an internal command for batch files. During batch execution, each REM command displays the comment stored on that command line in the batch file.

comment
is a string of characters (123 maximum) that you want displayed when batch execution encounters the REM command in the batch file. The only delimiters you can use in the comment are a blank space, tab, or comma:

```
REM This is a file to display the main directory
PAUSE Insert disk in drive B:
DIR B:
```

You can use a period (.) in place of REM to indicate a comment line. For the first line in the preceding example, you can enter:

```
.This is a file to display the main directory
```

Rename (REN)

REN [path]filename.ext filename.ext(cr)

REN is an internal command that renames files.

path
is the path through the subdirectory system to the directory containing the file you want to rename. Path always includes the drive name if the drive is not the default drive.

filename.ext
The first filename is the name of the file to be renamed. The second filename is the new name. The file itself remains on the diskette where it was originally created.

You can use wild-card characters in either filename. This command changes all the files with .DAT extension to .CMD extension:

ren *.dat *.cmd(cr)

The following command renames the file PHNLIST on drive B to PHNNUM. The file remains on drive B:

ren b:phnlist phnnum(cr)

If you rename a file with a name that is already in the directory, MS-DOS displays:

```
Duplicate filename or File not found.
```

PlusPC System Copy (RETROSYS)

RETROSYS source [/A] destination [/A](cr)

RETROSYS is an external command that copies the MS-DOS system files from one specified directory to another. If the destination is a diskette, the diskette must be in VICTOR format.

Because there is a unique set of files for I mode and one for V mode, RETROSYS can copy one or both sets of files. These files are as follows:

IDOS.SYS
VDOS.SYS
ICONFIG.SYS
VCONFIG.SYS
ICOMMAND.COM
VCOMMAND.COM
ICONFIG.BAT
VCONFIG.BAT

source
is the path to the directory from which you want to transfer the MS-DOS system files. If you specify only the drive letter, you must include the colon (A:). The source must contain the files ICOMMAND.COM, VCOMMAND.COM, and the operating system files. If you want to use the current default directory as the source, you can enter the following in place of the directory name:

(.)

destination
is the path to the directory to which you want to transfer the MS-DOS system files. If you specify only the drive letter, you must include the colon (A:). If you do not specify a destination drive letter, RETROSYS asks you for one.

/A
is the command switch that copies ICONFIG.SYS/VCONFIG.SYS and ICONFIG.BAT/VCONFIG.BAT to the destination. Any device drivers specified in ICONFIG.SYS or VCONFIG.SYS are also copied.

The following command copies all the system files from the current directory on drive A to the root directory of drive B:

retrosys a: b:(cr)

If you are currently in the root directory of drive A, you can enter the following command to accomplish the same system file transfer:

retrosys (,) b:(cr)

If you want to copy the files ICONFIG.SYS/VCONFIG.SYS and ICONFIG.BAT/VCONFIG.BAT from drive A to drive B, enter:

retrosys a:/a b:(cr)

Remove Directory (RMDIR)

RMDIR [path](cr)

RMDIR is an internal command that removes the last named subdirectory in a directory hierarchy. You can abbreviate RMDIR to RD. Chapter 3 gives more information about RMDIR and the subdirectory system.

path
is the path through the directory hierarchy to the subdirectory you want to delete.

RMDIR will not delete a subdirectory if it is not empty. If you try to delete the current directory or a subdirectory containing any entries other than the . and .. (notations indicating this directory level and the next higher level), MS-DOS displays:

```
Invalid path, not directory
or directory not empty
```

Delete (DEL) all files from the subdirectory you want to remove. Delete all directory entries from the subdirectory with the RMDIR command.

You can use RMDIR and the directory name without a backslash to delete a subdirectory entry in the current directory. Suppose you have a directory hierarchy of:

\MGR1.NAM\JOBS\BENEFITS

If you are currently working with the JOBS subdirectory, you can delete BENEFITS from JOBS by entering:

rmdir benefits(cr)

If you are using any directory except BENEFITS, the command:

rmdir \mgr1.nam\jobs\benefits(cr)

removes the BENEFITS subdirectory, if BENEFITS is empty. To remove the JOBS subdirectory, return to the root directory with the CHDIR command, make sure JOBS is empty, and then enter RMDIR with the path to JOBS.

Single Drive Disk Copy (SDCOPY)

(V mode only)

SDCOPY [drivename:](cr)

SDCOPY is an external command that copies the contents of one diskette onto a VICTOR format diskette in the same disk drive. You insert the source diskette (the diskette to be copied) into the drive, and its contents are read into memory. Then you remove the source diskette and insert a VICTOR formatted destination diskette into the same drive. The contents of the source diskette are written from memory onto the destination diskette. SDCOPY is useful for copying diskettes on a single-drive system. SDCOPY does not format the destination diskette; you must use a formatted diskette.

For more information on formatting a diskette in V mode, see the FORMAT command for V mode in this chapter.

drivename
is the name of the drive to be used to make the diskette copy.

When you issue the SDCOPY command, screen prompts ask you to insert the source diskette into the drive and then swap the destination diskette. You may have to reinsert the source and destination diskettes more than once depending upon the size of the diskette contents and the amount of available memory.

For example:

```
A>sdcopy(cr)
Single Drive Disk Copy Utility, Ver. x.y
Insert source disk, then press space bar
(return key to exit)(sp)
3 insertions of your source and destination
disks will be required.
Partial read #1 of 3
Insert formatted destination disk, then press
space bar (return key to exit)(sp)
Partial write #1 of 3
Insert source disk, then press space bar
(return key to exit)(sp)
Partial read #2 of 3
Insert formatted destination disk, then press
space bar (return key to exit)(sp)
Partial write #2 of 3
Insert source disk then press space bar (return
key to exit)(sp)
Partial read #3 of 3
Insert formatted destination disk, then press
space bar (return key to exit)(sp)
Partial write #3 of 3
SDCOPY completed
Insert source disk, then press space bar
(return key to exit)(cr)
A>_
```

You can stop the copy process by pressing the Return key in response to a prompt for a diskette. If you stop the copy before it is complete, the destination diskette is unusable.

You can use single- or double-sided diskettes. If you copy a single-sided diskette onto a double-sided diskette, the double-sided diskette becomes single-sided.

Search Files (SEARCH)

SEARCH files or directories [matching constraints] [actions](cr)

SEARCH.EXE is an external command that does these three things:

- Searches through the specified files and/or directories
- Selects matching files according to the specified constraints
- Either lists the matching file(s) or performs the specified action(s) on the matching file(s)

You specify the action you want SEARCH to perform with command switches (such as /CP for copy files). With SEARCH you can:

- Move, copy, or delete files
- Archive files by creating a disk archive file
- Change file attributes (such as read-only or read/write status)
- Find the sum of file sizes
- Delete empty subdirectories

Additional switches allow you to request prompting, exclude subdirectories from the search, and copy entire tree structures (subdirectory hierarchies).

Because SEARCH has several types of parameters, the syntax description for this utility is divided into sections listing and describing each type of parameter: *Files or Directories*, *Matching Constraints*, *Actions*, and *Action and Constraint Modifiers.*

The first section presents the syntax for "files or directories" and gives examples showing how to use file and directory names with SEARCH. In the other three sections, the command switches for that set of parameters are presented in table form. Examples showing all the SEARCH command parameters are given in a final section, *Hints for Using SEARCH.*

Files or Directories—The Domain

The first set of SEARCH parameters—files or directories—immediately follows the keyword SEARCH in a command. This parameter list consists of one or more subdirectory names and/or filenames which can include the MS-DOS wild-card characters. The list is the possible domain from which SEARCH selects files to operate on. If only a drive name is given, SEARCH begins at the current working directory of that drive and searches downward through any subdirectories below. To display the names of the current working directories, enter CD /A.

If you want SEARCH always to start at the root directory in the domain, use a \ at the beginning the path. For example, if you specify:

```
search a:\ b:\ other parameters(cr)
```

the SEARCH utility searches downward from the root directories on drives A and B only. Without the \, the search is relative; in other words, the search starts at the current working directory, which might not be the root directory.

You can mix filenames and directories to specify a domain of some specific files and the files in some subdirectories. This command:

```
search b:\rtc tod.src tod.asm other parameters(cr)
```

considers the file or directory B:\RTC, and any subdirectories below it, and the two TOD files on the default drive when executing the other parameters.

Matching Constraints

The second set of (optional) parameters in a SEARCH command is matching constraints. These are command switches that further define which files in the specified domain of files and/or directories you want to affect with SEARCH. Only files from the domain that meet all the conditions listed as matching constraints are acted on. You can only use each of the switches listed in Table 6-1 once in a command line.

Table 6-1: Matching Constraint Switches for SEARCH

SWITCH	EFFECT
/NAME filenames	Selects files from the domain that match the filenames following /NAME. To list more than one filename, separate them with exclamation marks, like this: **/name rtc.asm!tod.asm!tod.obj(cr)** Details about including filenames with SEARCH are given in the paragraphs following this table.
/^NAME filenames	Selects files from the domain that do NOT match any of the names following /^NAME. See the examples following this table.
/FNAME pathname	Selects files whose full pathnames, including subdirectories and filename, match the names following /FNAME. See the examples following this table for a description of what the pathnames can be.
/^FNAME pathname	Selects files whose full pathnames do not match any of the names following /^FNAME. See the examples following this table.
/ATTR attribute expression	Selects files whose attributes make the specified attribute expression true. The expression consists of letters representing one of the five "file attributes" described here. Multiple letters can be separated with & (for AND), ! (for OR), and ^ (for unary NOT). You can also use parentheses for grouping subexpressions. The five file attribute letters are: A = archive status is set on (i.e., the file has been edited since the last archive) D = this "file" is a subdirectory H = this file is a hidden file R = this file is read-only S = this file is a system file To select read/write files (files that are not read-only) with archive status set on: **search a:\ /attr a&^r(cr)** To display the status of the five file attributes for your files, use the LS /L command (or SEARCH with the /LS action switch).

SWITCH	EFFECT
/COMP directoryname	Compares the files in the domain with files of the same name in the specified subdirectory. Selects only those files that have the same name and are equivalent to a file in the specified directory. Any differences between the two files are displayed. Include the /T switch with /COMP to preserve the directory tree structure of the domain. With /T, the target directory name for the compare is computed as follows: 1. The drive prefix (and any .\ or ..\ prefixes) are deleted from the directory name of the file found (i.e., the drive name is ignored). 2. The remainder of the directory name including the filename from the domain is appended to the directory named after the /COMP to produce the file which is to be compared. For example, SEARCH A: /COMP TEST /T compares files on drive A with files in the relative path TEST. If the file A:\X\Y exists, SEARCH compares it to TEST\X\Y but not to TEST\Y. SEARCH A: /COMP TEST without the /T switch compares A:\X\Y to TEST\Y.
/^COMP directoryname	Selects files from the domain if there is not a file of the same name in the specified path OR (if there is a file of the same name) the two files are not equivalent. Any differences between the two files are displayed. The description of /COMP gives more details.
/EXISTS directoryname	Selects files from the domain whose names match filenames in the specified subdirectory. If /T is used, the path is computed as for /COMP.
/^EXISTS directoryname	Selects files from the domain whose names do not match any filenames in the specified path. If /T is included, the path is computed as for /COMP.
/LE filename	Selects files from the domain whose time of creation (date and time) precedes or is the same as that of the named file. Read LE as "less than or equal to."
/LT filename	Selects files whose time of creation (date and time) precedes that of the named file. Read LT as "less than."
/GE filename	Selects files whose time of creation (date and time) follows or is the same as that of the named file. Read GE as "greater than or equal to."

SWITCH	EFFECT
/GT filename	Selects files whose time of creation (date and time) follows that of the named file. Read GT as "greater than."
/LEDATE filename	Selects files whose date of creation (regardless of time) precedes or is the same as that of the named file. Read LE as "less than or equal to."
/LTDATE filename	Selects files whose date of creation (regardless of time) precedes that of the named file. Read LT as "less than."
/GEDATE filename	Selects files whose date of creation (regardless of time) follows or is the same as that of the named file. Read GE as "greater than or equal to."
/GTDATE filename	Selects files whose date of creation (regardless of time) follows that of the named file. Read GT as "greater than."
/SIZE n	Selects files that are at least *n* bytes in size.
/SIZEB n	Selects files that are at least *n* (512-byte) blocks in size.

File and directory names listed with the /NAME, /^NAME, /FNAME, and /^FNAME switches can have several types of wild-card characters, some of which are themselves switches. The wild cards you can use are as follows:

- The MS-DOS ? wild card matches any single character, or none.
- The * matches any number of characters anywhere in the text. In most MS-DOS commands, *A* is equivalent to * because the first * pads to the end of the name. With the /NAME switch, however, *A* matches any text that contains an A.
- The /. switch makes wild cards also match the file extension separator (the . between the filename and extension). For example, without the /. switch, you must specify *.* to match all files; using * with the /. switch matches all files.
- The /\ switch makes wild cards also match the \ in a pathname. For example, A**.PAS matches A\X\Y.PAS and A\Z\W.PAS, but not A\X\Y\Z\Q\R\X.PAS. Using the * with the /\ switch also matches this last path.

- Multiple-choice selection. If you enclose characters in brackets, SEARCH looks for only the characters listed. For example, [AEIOU]*.ASM matches any filename beginning with a vowel and having the extension .ASM.

For example, these two /NAME or /^NAME phrases are equivalent:

```
[bv][tc][1k]fd*.*(cr)
[bv][tc][1k]FD* /.(cr)
```

Both match any file with B or V as its first character, T or C as its second, 1 or K as its third, and FD as its fourth and fifth characters. The second example uses the /. switch to match the . between the name and extension. Note that upper- and lowercase do not matter.

The next example matches any file in the root directory on drive A or in any subdirectory on drive A which does not have the extension .ASM:

```
search a:\ /^name *.asm(cr)
```


The next example matches any file in drive A's root directory or any subdirectory on A whose name does not begin with B or V:

```
search a:\ /^name [bv]*.*(cr)
```

The last example uses both /NAME and /^NAME. The files selected from the domain, which is the root directory of A, must have the extension .ASM but cannot begin with BT1:

```
search a:\ /name *.asm /^name BT1*.*(cr)
```

The difference between the /FNAME and /NAME switches is that /NAME performs the matching test only on the file's name, while /FNAME performs its test against the full pathname for that file, starting at the drive name and the root directory.

For example, the following command selects filenames beginning with SRC and having no filename extension in the root directory on drive B, or any subdirectory on drive B that begins with SRC.

```
search b:\ /fname b:\src*(cr)
```

Adding the /\ switch matches any subdirectories below the first one. Adding /. matches any file extension. For example, this command matches files in any path beginning with B on drive A:

```
search a:\ /fname a:\b* /\ /.(cr)
```

To match drive designators, the format is D:\. For example:

```
search a:\ /fname a:\t* /\ /.(cr)
```

selects files from the root directory on drive A that begin with T, or any files in subdirectories on A whose directory names begin with T.

The last example selects all files from drive A that are not in subdirectory A:\TEMP:

```
search a:\ /^fname a:\temp\* /\ /.(cr)
```


Actions

The third set of parameters for SEARCH are action switches. These options specify the function to be performed on the files that meet the matching constraints. With action switches you can set and reset file attributes; move, copy, delete, and back up files; print directory entries; and find the sum of the file sizes.

Table 6-2 lists and describes the switch settings for actions. Any action marked with an asterisk can be specified only once in a SEARCH command.

Table 6-2: Action Switches for SEARCH

SWITCH	EFFECT
/ATTR + attrs	Sets the specified file attributes on. Specify "attrs" as single-character attributes, as for the /ATTR matching constraint (see Table 6-1). Do not use separators between attribute letters. For example, this command sets archive and read-only status on: **/attr + ar(cr)**
/ATTR- attrs	Set the specified attributes off.
/BACKUP directoryname *	Is an abbreviation for: **/cp directoryname /attr a /attr- a(cr)** That is, this switch finds archivable files (those with archive status) and resets archive status off after the copy. To preserve the tree structure of the files you back up, use the /T switch. You cannot use /BACKUP with /CP.
/CP directoryname *	Copies the matching files into the specified directory. This cannot be used with /BACKUP. You can use the /T option as for the /COMP matching constraint switch (see Table 6-1).
/LS	Lists a directory of matching files in long format. /LS displays file attribute settings, file size, time and date of creation, and the full pathname. If an attribute is not set for a file, / is printed; if it is set, the attribute is printed. For read/write files, however, w is printed instead of the /. See /ATTR in Table 6-1 for a list of the attributes.
/MV directoryname *	Moves the matching files from the domain into the specified subdirectory. The original files are deleted. If the /T option is specified, the tree structure of the source file is preserved. For example, this command: **search a: /mv z /t(cr)** finds A:\X\Y; the file Z\X\Y would be created rather than Z\Y (the file created without /T).

SWITCH	EFFECT
	The destination directory name is formed from the source pathname as follows: 1. The drive prefix and any .\ or ..\ is removed from the source directory name. 2. The remainder of the source directory name is appended to the directory name after the specified /MV directory name.
/PRINT	Lists the matching files from the domain. PRINT is the default action for SEARCH.
/RM	Removes (deletes) the matching files. If you specify both /CP and /RM, the copy is done before the removal. If you specify /MV with /RM, /RM has no effect.
/RMDIR	Removes any empty directories encountered. If /MV is specified, the directory remove is attempted after the files are moved. This command moves all of X to drive B, and then removes all of subdirectory X: **search x /mv b: /rmdir(cr)**
/SUM	Displays the sum of the sizes of matching files. This is the exact size of the data in the files and does not consider allocation units' round-off.
/TARA filename.ext *	Appends the matching files to an existing TAR (tape archive) file. Otherwise acts same as /TARC.
/TARC filename.ext *	Creates a TAR (tape archive) file named FILENAME.EXT containing all matching files. For example, this command puts all .PAS files on the default drive into the file PASBKUP.TAR: **search *.pas /tarc pasbkup.tar(cr)**
/TARX filename.ext *	Extracts the matching files from the tape archive file FILENAME.EXT. The only action you can perform with this option is to /LS, /PRINT, /SUM, or /CP files. The /RM or /MV switches are ignored. This command lists the files in the tape archive file PASBKUP.TAR: **search /tarx pasbkup.tar /ls(cr)** This command copies the files in the given TAR file into directory XYZ on the default drive: **search /tarx pasbkup.tar /cp xyz(cr)**

*These switches can appear only once in a SEARCH command.

Action and Constraint Modifiers

You can use the switches listed in Table 6-3 to modify both the matching constraints and the actions listed in a SEARCH command. You can group single-letter options after a single switch signal (such as /AV) providing they do not form one of the multiple-letter options. Each multiple-letter option must have a separate signal. For example, you cannot combine the /AV switch with the /NR switch to make /NRAV.

Table 6-3: SEARCH Action and Constraint Modifiers

SWITCH	EFFECT
/A	Matches only files with the archive status on. Short for /ATTR A.
/N	Takes no actions. SEARCH displays what it would have done. The /N switch is equivalent to using /P and always responding negatively.
/NR	Does no recursive searching of subdirectories. SEARCH normally searches any directories you specify and all subdirectories. Using /NR limits searches to only the directories specified.
/P	Prompts (asks yes/no) for /CP, /MV, /RM, /RMDIR, /TARA, /TARC, /TARX, /BACKUP, /ATTR-, or /ATTR + options. Use this switch to verify SEARCH's actions before you take an action that would have a permanent or destructive effect.
/S	Is silent (displays nothing).
/T	Preserves tree structure of directories for /MV, /CP, /COMP, /^COMP, /EXISTS, /^EXISTS, /TARX, and /BACKUP options.
/V	Preserves volume/drive structure. An option for /TARX only, /V returns files from tape archive to their original volume or drive.
/.	Allows the file extension separator (.) to be wild-carded.
/\	Allows the directory separator (\) to be wild-carded.

Hints for Using SEARCH

- Remember that all matching constraints must be met (they are ANDed together) before a file is processed (listed, deleted, moved, or archived) by SEARCH.
- You can mix and match any constraints, action switches, and modifying options switches.
- If you are doing anything that might cause data to be lost or reformatted, add the /P switch to prompt you for verification of the action, or append the /N switch to display the actions that would occur if you execute the command.

Use the following list of examples as guidelines when you use the SEARCH command:

1. If you specify only a drive name in the domain, such as:

   search f: /name *.pas(cr)

   SEARCH does not necessarily search all files on that drive. SEARCH starts at the current working directory and skips any higher directories. To search drive F for files with the .PAS extension, either do a CD F:\ before the preceding command, or use:

   search f:\ /name *.pas(cr)

2. To list all the directories on drive D, type:

   search d:\ /attr d(cr)

3. To copy all files that have been modified since the last time you did a backup, use:

   search a:\ /backup b:(cr)

   This copies modified files to drive B and resets their archive status off.

If the archive status of your files is unknown, enter:

```
search a:\ /attr+ a(cr)
```

to set archive status on for all files (so all files will be archived).

4. Because some commands are abbreviations, remember that there are some restrictions. For example,

```
search a: /backup b: /attr r(cr)
```

is illegal because /BACKUP B: is an abbreviation for /CP B: /ATTR A /ATTR- A, and only one /ATTR is allowed. Instead you should enter:

```
search a: /cp b: /attr a&r /attr- a(cr)
```

5. To find all files created on the same day as SOFTWARE.MEM, on drives A and C, use:

```
search a:\ c:\ /ledate software.mem /gedate software.mem(cr)
```


This command lists all files with dates less than or equal to AND greater than or equal to SOFTWARE.MEM's date. Thus, all files created on that date will be listed.

To copy all of these files to drive B's working directory, enter:

```
search a:\ c:\ /ledate software.mem /gedate software.mem
  /cp b:(cr)
```

6. To list all hidden and system files on drives/volumes A, C, D, and E, enter:

```
search a:\ c:\ d:\ e:\ /attr s!h /ls(cr)
```

7. Before you archive files with SEARCH /TARC, you can see the amount of space that will be occupied by the archived files. If you are about to archive drives/volumes A, C, and D, enter:

```
search a:\ c:\ d:\ /a /sum(cr)
```

The results tell you the size of the tape archive file that will be created when these files are archived.

8. To archive the files from drives/volumes A, C, and D, enter:

```
search a:\ c:\ d:\ /a /tarc b:backup /attr- a(cr)
```

This command finds all files with archive status on (/A), archives the files into a single file called BACKUP on drive B (/TARC B:BACKUP), and then resets archive status off (ATTR- A).

9. To re-arrange your directory structure, you can use SEARCH to copy entire subdirectories. For example, to move the subdirectory A:\COMMAND\SOURCES and all of its subdirectory structures, including all files, to C, use:

```
search a:\command\sources /mv c:\ /t(cr)
```

In this example, using the /T option preserves the tree structure (\COMMAND\SOURCES\...). Without the /T switch, all the files of all the directories would be copied into the current working directory of C.

10. To remove any empty directories on drives E and F, use:

```
search e:\ f:\ /rmdir(cr)
```

The /RMDIR switch can be used with any other actions. Only empty directories will be removed; also, any drive's current working directory cannot be removed.

11. If there are duplicate files in the two directories F:\PROJECTS and G:\MYPROJ, you can delete the redundant files from G:\MYPROJ with this command:

```
search g:\myproj /exists f:\projects /rm(cr)
```

If you want to delete only files that contain the same data as the similarly named files in F:\PROJECTS, use the following:

```
search g:\myproj /comp f:\projects /rm(cr)
```

Set Equivalent Value (SET)

SET [string1 = [string2]](cr)

SET is an internal command that sets one string value equivalent to another string. In subsequent MS-DOS or application program operations, the equivalent value is substituted for the first named string.

With SET, you change the operating environment. The operating environment is a series of names and parameters built by the command processor and passed to all programs that it invokes. You can SET almost anything into the environment. The only syntax requirement is that a single equals sign (=) be in the command line.

Entering SET without any parameters displays all the values that have been set with the SET command. For example:

set(cr)

lists all the values currently set, such as:

```
path=
prompt=$n$g
```

string1
is an environment-name or variable name that you want to set equivalent to the value of string2. For example, you can duplicate the function of the PROMPT command with SET. Enter:

set prompt = YES?(cr)

to change the system prompt to YES?.

To clear any setting, enter the SET command giving only the first string and an equals sign:

```
set prompt = (cr)
```

string2
is the value that you want to substitute for string1 in subsequent operations.

SET is useful if you are doing batch file processing or using an application program that repeats a variable. For example, in a batch file you can create replaceable parameters with names instead of the usual numbers. Replaceable parameters defined as names must have a leading and trailing %, such as %FILE%. You can assign the value to use for that parameter with the SET command:

```
set file = domore(cr)
```

When MS-DOS processes the batch file, %FILE% becomes the filename DOMORE. Using this procedure, you do not have to edit a file to change the parameter names before you run the batch file.

Shift Replaceable Parameters (SHIFT)

SHIFT(cr)

SHIFT is an internal command for batch files that moves all replaceable parameters one position to the left:

%1 replaces %0
.
.
.
%10 replaces %9

When there are more than 10 parameters on a command line, each additional parameter shifts into %9.

For example, this batch file named DIRLOOK.BAT uses the SHIFT command to display as many directories as you enter when you load the batch file:


```
dir /p %1
:nexdir
shift
if %1 == stop goto lastone
dir /p %1
goto nexdir
:lastone
cd \
```

If you specify:

dirlook managers scheds stop(cr)

MS-DOS searches the root directory for the MANAGERS and the SCHEDS directories. During batch processing, MS-DOS displays each batch command (except REM lines or lines beginning with colons) and then performs the command on the line.

MANAGERS is used for %1. Because of the SHIFT command, %1 becomes SCHEDS. PROJ then replaces SCHEDS, and so on. The batch file completes when STOP equals STOP in the IF command. MS-DOS displays:

```
A>dir /p managers
```

Next the MANAGERS directory is displayed.

```
A>shift
A>if scheds == stop goto lastone
A>dir /p scheds
```

Next the SCHEDS directory is displayed.


```
A>goto nexdir
A>shift
A>if stop == stop goto lastone
A>cd \
```

The CD \ at the end of the file returns you to the root directory. See Chapter 3 for information on changing directories, or see CHDIR in this chapter.

Sort Data (SORT)

[command |] SORT [/R] [/+n] [< [drivename:]input source]
[> [drivename:]output source](cr)

SORT.EXE is an external command that filters data by rearranging it alphanumerically (using the ASCII collating sequence—see Appendix A). The data to be rearranged can be the output from an MS-DOS command.

The command can be any external MS-DOS command whose output you want to sort. The | symbol tells MS-DOS to direct (pipe) the output from the MS-DOS command to the SORT command, described in Chapter 5.3.

For example, this command sorts the current directory alphabetically:

dir | sort(cr)

The command:

dir | find "< DIR >" | sort(cr)

uses the output from the DIR command as input to the FIND command. The output of FIND, the names of all the subdirectories in the current directory, is sorted alphabetically and displayed on the screen.

/R
tells MS-DOS to reverse the alphanumerical sort (Z to A, or *n* to 0 where *n* is an integer).

/+n
tells MS-DOS to sort the data by columns. *n* is the number of the column where the SORT is to start. For example, this command:

dir | sort /+1(cr)

sorts the directory alphanumerically by filename, which is the first column in the directory display.

This command does a reverse alphanumeric sort of the directory:

```
dir | sort /r /+1(cr)
```

<

tells MS-DOS to use the parameters following the < symbol as input.

drivename

is the name of the drive containing either the input data to be sorted or the sorted output data. The drive name is optional when the input or output source is on the default drive.

input source

is the data you want to sort. It can be a disk file or data from a MODEM. For a disk file, specify the name of the file and the path to the file if it is not in the current directory. For a MODEM or any other device, you must specify the physical device name.

>

tells MS-DOS to use the parameters following the > symbol as output. You must include an output source if you include the > symbol. For example, if you enter:

```
dir | sort >(cr)
```

MS-DOS displays:

```
File creation error
```

output source

is the destination of the sorted data from the input source. It can be files, printers, or other devices. For files you must include the path to the file unless it is in the current directory. This command:

dir | sort > lst(cr)

sorts the directory alphabetically and sends the output to the LST device, the printer. The default output is > CON. If you do not include an output source, the output from the SORT command is sent to CON, the CRT screen.

The command:

find "95066" mail.lst | sort > zip.cod(cr)

pulls out all the records containing the zipcode 95066 from the MAIL.LST file and sorts them alphabetically into the ZIP.COD file. ZIP.COD is created by this command, but if ZIP.COD already exists, its contents are overwritten by the output of SORT. To see the sorted data records in ZIP.COD, enter:

type zip.cod(cr)

MS-DOS displays this message, followed by an alphabetical list of each 95066 record:

```
---------- mail.lst
```

Print End of File (TAIL)

TAIL [/n] [files or directories](cr)

TAIL.EXE is an external command that displays the last n lines of a file or list of files.

/n
is the number of lines at the end of the file(s) you want to display onscreen. If you do not include a number, TAIL prints the last 10 lines. The maximum value is approximately 450. If a line exceeds 130 characters, TAIL truncates it.

files or directories
is a list of subdirectory names and/or filenames to be operated upon. If you do not include files or directories, TAIL gets its input from the keyboard or from redirected input.

For example, this command lists the last 3 lines of every file with .LST extension in the current directory:

tail /3 *.lst(cr)

Either of the next two commands lists the last 10 lines of the file ACCOUNTS. The < symbol in the first command indicates that ACCOUNTS is being redirected as input to the TAIL command.

tail < accounts(cr)

tail accounts(cr)

The following command prints the last 50 lines of every file in the \TEMP subdirectory on drive A:

tail /50 a:\temp(cr)

TIME

TIME [hh][:mm][:ss](cr)

TIME is an internal command that sets the time for the operating system. If you enter the command without parameters, the current time is displayed and you are prompted for a new time. Press the Return key if you do not want to change the time. For example:

```
A>time(cr)
Current time is 13:01:10.00
Enter new time:9:05(cr)
```

The new time known to the operating system is now 09:05:00.00. The colon separates the hours, minutes, and seconds. The period separates the hundredths of a second.

hh
represents the hour. Use 00 to 23.

mm
represents the minutes. Use 00 to 59.

ss
represents the seconds. Use 00 to 59. You do not have to type the seconds or hundredths of a second.

If you enter the TIME command with valid parameters (numbers, not letters, in the indicated range for each parameter), MS-DOS displays the system prompt.

If the parameters are invalid, such as letters instead of numbers or a larger than acceptable range of numbers, MS-DOS displays:

```
Invalid time
Enter new time:_
```

MS-DOS also displays this message if you use invalid delimiters, such as a hyphen instead of a colon or a period.

List Directory Paths (TREE)

TREE [drivename:] [/F](cr)

TREE is an external command that lists all directory paths found on the specified drive. TREE can also list all the files in each directory. If you do not specify a drive letter with the TREE command, the default drive is used.

drivename:
is the name of the drive for which you want to list all the directory paths.

/F
lists the names of the files in each subdirectory.

If you enter the TREE command without any parameters, you might see a listing on your screen similar to the following:

```
C>tree(cr)
DIRECTORY PATHS FOR VOLUME ..... VOLUMENAME

Path:     C:\CAM
Sub-directories:  -none-

Path:     C:\TEMP
Sub-directories:  -none-

Path:     C:\LC
Sub-directories:  VICTOR
                  SOURCE
```

If you enter the TREE command with the /F option, the preceding listing is expanded to include all the files in each subdirectory.

```
C>tree/f(cr)
DIRECTORY PATHS FOR VOLUME ..... VOLUMENAME

Path:     C:\CAM
Sub-directories:  -none-

     Files:       CHAPT1.TXT
                  CHAPT2.TXT
                  CHAPT3.TXT
                  APPA.TXT
                  APPB.TXT

Path:     C:\TEMP
Sub-directories:  -none-

     Files:       LTRFORM.LIB
                  MEMOFORM.LIB
                  CALVIN.LTR

Path:     C:\LC
Sub-directories:  VICTOR
                  SOURCE

     Files:       VICTOR.BAT
                  MAIN.OBJ
                  HOMER.BAT
```

Type File (TYPE)

TYPE [path]filename.ext(cr)

TYPE is an internal command that displays the contents of text files. You cannot use TYPE with .COM or .EXE files because they contain non-alphanumeric characters, such as Control or Escape sequences. Tabs expand to 8 characters (column 8, 16, and so on through column 80).

path
is the path through the subdirectory system to the file you want to display.

filename.ext
is the name of the file to be displayed.

For example, to display the contents of the JOBS.LST file in the \MGR1.NAM\JOBS subdirectory, enter:

type \mgr1.nam\jobs\jobs.lst(cr)

Reload Command Processor (VCOMMAND)

VCOMMAND is an external command that invokes the command processor, VCOMMAND.COM. See the description of ICOMMAND in this chapter.

Version of (VER)

VER(cr)

VER is an internal command that displays the version of MS-DOS that you are using. For example, the following command:

ver(cr)

displays:

```
MS-DOS Version  x.y
```

where *x.y* is the version number of the operating system.

Verify Write to Disk (VERIFY)

VERIFY [ON](cr)

or

VERIFY [OFF](cr)

VERIFY is an internal command that verifies whether subsequent data written by MS-DOS onto the disk can be read by MS-DOS when requested. VERIFY has the same function as the V switch on the COPY command. VERIFY without parameters displays the current setting (ON or OFF) of the command. For example, if you type:

verify(cr)

MS-DOS displays:

```
VERIFY is xx
```


where *xx* is "on" or "off."

ON
: tells MS-DOS to verify the data each time it writes data to the disk. MS-DOS issues an error message if it cannot do the write operation. Note that when VERIFY is ON, any command involving a write to the disk performs slower because of the extra time involved to verify the operation.

OFF
: turns the verify instruction off.

Volume Label Display (VOL)

VOL [drivename:][/C](cr)

VOL is an internal command that displays the Volume ID of a diskette or fixed (hard) disk volume. A Volume ID is a name or label you give to a diskette during formatting, or to a fixed (hard) disk volume during initialization. A Volume ID can be up to 11 characters long, using the valid characters for filenames. MS-DOS displays the Volume ID for a diskette or volume when you enter a DIR command, a CHKDSK command, or a VOL command.

drivename
is the name of the drive whose ID you want to check. If you do not specify a drive, MS-DOS displays the ID for the default drive.

/C
is a switch you use to request a change in the Volume ID.

If you try to display the Volume ID for a diskette or volume that does not have an ID, MS-DOS displays:

```
Volume ID for drive x: has not been set.
```

where *x* is the name of the drive.

If you type:

vol b: /c(cr)

MS-DOS displays the Volume ID for drive B, and prompts you for a new Volume ID:

```
Volume ID for drive B: is ARCHIVE1031
Enter new Volume ID:_
```

After you enter a new Volume ID and press Return, MS-DOS writes the new ID to the disk. You can display the new Volume ID for drive B by entering:

vol b:(cr)

6

Word and Line Count (WC)

WC [/R] [/W](cr)

or

WC [/R] [/W] [path...](cr)

or

WC [/R] [/W] [filename...](cr)

WC.EXE is an external command that counts words and/or lines in a file or files. WC can count all the files in a directory or in a hierarchy of directories. WC lists the number of lines, the number of words, the filename, and the total count of lines and words. Entering WC with no parameters counts the lines and words in the keyboard input (or redirected input) that follows, up to end-of-file (CTRL-Z Return).

/R

is a command switch that expands the count to include files in all subdirectories below the directory named in the WC command. To use /R, you must list a directory, not a file, in your WC command.

The following example uses /R with the root directory to count the words and lines in all the files in every directory on the default drive:

```
A>wc /r \(cr)
      21 L      386 W  \VMS-DOS.SYS
      43 L      495 W  \VCOMMAND.COM
       5 L       16 W  \VCONFIG.SYS
      49 L      297 W  \SUBDIR1\FINAL.MEM
      86 L      455 W  \SUBDIR2\CHAP2.DOC
       . .        . .    .   .   .
       . .        . .    .   .   .
     563 L     6511 W  total
```

/W

is a command switch that omits the word count from the WC operation. Counting only lines is faster than counting words and lines.

path

is the path through the subdirectory system to the file or subdirectory you want to count. If you list a directory name (or the single \ to indicate the root directory) without a filename, WC counts all the files in that directory.

filename

is the name of the file you want to count. You can use wild-card characters to refer to more than one file.

The first example below counts lines and words in the FINAL.MEM file, and the second example counts lines in the FINAL.MEM and FINAL.BAK files:

```
wc final.mem(cr)

wc /w final.mem final.bak(cr)
```

You can list multiple files and directories in a WC command, and the files and directories can be on different drives. The following command counts the number of lines in all files with .DAT extension on the default drive as well as all files in the subdirectory \DRAFT1 on drive B:

```
wc /w *.dat b:\draft1(cr)
```